



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST-NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/944,721	08/31/2001	Dennis A. Quan JR.	POU920010121US1	1719

23334 7590 10/19/2006

FLEIT, KAIN, GIBBONS, GUTMAN, BONGINI  
& BIANCO P.L.  
ONE BOCA COMMERCE CENTER  
551 NORTHWEST 77TH STREET, SUITE 111  
BOCA RATON, FL 33487

EXAMINER
----------

CHUONG, TRUC T

ART UNIT	PAPER NUMBER
----------	--------------

2179

DATE MAILED: 10/19/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b> 09/944,721	<b>Applicant(s)</b> QUAN, DENNIS A.	
	<b>Examiner</b> Truc T. Chuong	<b>Art Unit</b> 2179	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 07 August 2006.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213:

**Disposition of Claims**

- 4) ☒ Claim(s) 1-23 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-23 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)                     | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____                                      |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)          | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____  | 6) <input type="checkbox"/> Other: _____                          |

### **DETAILED ACTION**

This communication is responsive to an amendment, filed 08/07/06. Claims 1-23 are pending in this communication, and this action is made final.

The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior office action.

#### ***Claim Rejections - 35 USC § 102***

1. Claims 1-23 are rejected under 35 U.S.C. 102(b) as being anticipated by Microsoft Explore Screen Captures ("Microsoft", figures 1-5, and Window NT 4 Workstation, figures 6-9 to verify the features of Microsoft Explore have been available since 1996).

From Microsoft Windows Operating System (regular versions or NT versions), selecting → Microsoft Explore (fig. 8) to open this window → it displays a window with two panes (figs. 1-4), which contain folders, documents, programs, applications, software, etc. → different groups of the folders associate with an appropriate menu for that group (figs. 2-4 show different menu options or some of the options is grey out) → it means associative array(s) contain(s) global context flags to tell the OS the status of the programs/applications/folders, based on the given information, the OS will provide or turn on/off some of the necessary options to the user (figs. 2-5).

As to claim 1, Microsoft shows a method for presenting zero or more User Interface (UI) objects as pad of a UI on an information processing system, the method comprising:

associating an associative array (An associative array of Microsoft is an array of listing states causing the system whether to dim (gray out) or remove options from menu options, e.g.,

Art Unit: 2179

figs. 2-5) with one or more entries to each of a plurality of UI objects presentable as part of a UI (e.g., menu options, figs. 2-3);

presenting at least one UI object based on a state of at least one global context flag for the UI {figs. 2-4 show different menu options or some of the options is grey out. It means associative array(s) contain(s) global context flags to tell the OS the status of the programs/applications/folders, based on the given information, the OS will provide or turn on/off some of the necessary options to the user (figs. 2-5)};

receiving at least one of a response from an end-user to the presentation of the at least one UI object and an event-based trigger (the user make selections and interact with the menu items, e.g., figs. 2-5);

altering the state of the global context flag based on the response from the end-user (the user make selections and interact with the menu items; based on user selections, figs. 2-4 show different menu options or some of the options is grey out);

performing a Boolean comparison between the global context flag and one or more of the entries in the associative array for each of the UI objects {the associative array(s) contain(s) global context flags to tell the OS the status of the programs/applications/folders, based on the given information, the OS will provide or turn on/off some of the necessary options to the user (figs. 2-5)}; and

presenting zero or more of the plurality of UI objects as part of the UI to the end-user based upon a result of the Boolean comparison (e.g., figs. 2-5).

As to claim 2, Microsoft shows the method wherein the step of presenting to an end-user the at least one UI object further comprising at least one of the following:

Art Unit: 2179

presenting a visual image (graphical menu, e.g., figs. 3-4);

creating a vibration;

playing a sound;

emitting a scent; and

emitting a taste.

As to claim 3, Microsoft shows the method wherein the step of associating an associative array with one or more entries to each of a plurality of UI objects that may be presented as part of a UI includes associating an associative array with zero or more entries to each of a plurality of UI objects that may be presented as part of the UI by retrieving the associative array from a file when an application using the UI is loading (e.g., figs. 2-5).

As to claim 4, Microsoft shows the method wherein the global context flags are changed after the application using the UI receives any input from:

an end-user comprising an input from a group of input devices consisting of a keyboard, mouse, pointing device, digitizing pen, light pen, track ball, touch screen, motion detector, chemical sensor, sound sensor, and eye movement sensor (Microsoft Windows can run on a regular PC which contains keyboard and mouse, etc.).

As to claim 5, Microsoft shows the method wherein the global context flags are changed after the application using the UI receives an event-based trigger from one or the following events:

a time event comprising an input from a time of day or elapsed time (Microsoft inherently shows this feature because the invention of Microsoft utilizes a regular IBM computer

Art Unit: 2179

with Print command of fig. 4 is also available on the menu; it means that the Print command contains well known features such as: a file status, printing status, date, time, etc.); and

an information processing system event comprising a file status, a printing status, a modem status or a power supply status (the Print command contains well known features such as: a file status, printing status, date, time, etc.).

As to claim 6, Microsoft shows the method wherein the step of performing a Boolean operation is selected from the group of Boolean operations consisting of “AND”, “OR” and “XOR” {the system will compare the state of user selection, and the associative array(s) contain(s) global context flags to tell the OS the status of the programs/applications/folders, based on the given information, the OS will provide or turn on/off some of the necessary options to the user (figs. 2-5)}.

As to claim 7, Microsoft shows the method wherein the step of associating one or more attributes to each of a plurality of UI objects from a table so that any changes in the table can be made without recompiling and/or re-linking an application using the UI (menus options of figs. 2-5 are also tables that the items can be changed/removed/deleted without recompiling the application).

As to claims 8-10, they are the equivalent system claims of method claims 1, 4, and 2 respectively and are rejected under a similar rationale.

As to claims 11-17, they are the equivalent program product claims of method claims 1-7 respectively and are rejected under a similar rationale.

As to claim 18, Microsoft shows a method for presenting User Interface (UI) objects as pad of a UI on an information processing system, the method comprising:

Art Unit: 2179

creating a first associative array for controlling a property of at least one UI object in a UI {different groups of the folders associate with an appropriate menu for that group (figs. 2-4 show different menu options or some of the options is grey out) → it means associative array(s) contain(s) global context flags to tell the OS the status of the programs/applications/folders, based on the given information, the OS will provide or turn on/off some of the necessary options to the user (figs. 2-5)};

performing a Boolean comparison between:

at least one global context flag in the first associative array (An associative array of Microsoft is an array of listing states causing the system whether to dim (grey out) or remove options from menu options, e.g., figs. 2-5, and the different menu options between figs. 3 and 4); and

at least one entry in a second associative array of at least one UI object graphical selections associated with the at least one UI object (the different menu options between figs. 3 and 4); and

presenting updates to the UI objects as past of the UI to an end-user in response to the Boolean comparison {the associative array(s) contain(s) global context flags to tell the OS the status of the programs/applications/folders, based on the given information, the OS will provide or turn on/off some of the necessary options to the user (figs. 2-5)},

wherein the first associative array and the second associative array are indexed by a string (associative arrays as explained above, e.g., figs. 2-5).

As to claim 19, Microsoft shows the method further comprising:

receiving at least one of:

Art Unit: 2179

a response from the end-user to the updates to the UI objects (figs. 2-4 show different menu options or some of the options is grey out. It means associative array(s) contain(s) global context flags to tell the OS the status of the programs/applications/folders, based on the given information, the OS will provide or turn on/off some of the necessary options to the user (figs. 2-5)), and an event-based trigger.

As to claim 20, Microsoft shows the method further comprising:

changing a state of the global context flag based on the response received from the end-user (the user make selections and interact with the menu items; based on user selections, figs. 2-4 show different menu options or some of the options is grey out).

As to claims 21-23, they are the equivalent method claims of claims 4-6 respectively and are rejected under a similar rationale.

2. Claims 1-23 are rejected under 35 U.S.C. 102(e) as being anticipated by Giesen et al. (U.S. Patent No. 6,826,729 B1).

As to claim 1, Giesen shows a method for presenting zero or more User Interface (UI) objects as part of a UI on an information processing system, the method comprising:

associating an associative array (An associative array of Giesen is an array of listing states causing the system whether to dim (gray out) or remove options from menu options, e.g., figs. 4-5) with one or more entries to each of a plurality of UI objects presentable as part of a UI (e.g., menu options, figs. 4-9);

presenting at least one UI object based on a state of at least one global context flag for the UI {figs. 4-9 show different menu options or some of the options is grey out. It means



Art Unit: 2179

associative array(s) contain(s) global context flags to tell the system the status of the programs/applications/folders, based on the given information, the system will provide or turn on/off some of the necessary options to the user (figs. 4-9));

receiving at least one of a response from an end-user to the presentation of the at least one UI object and an event-based trigger (the user make selections and interact with the menu items, e.g., figs. 4-5);

altering the state of the global context flag based on the response from the end-user (the user make selections and interact with the menu items; based on user selections, figs. 4-5 show different menu options or some of the options is removed);

performing a Boolean comparison between the global context flag and one or more of the entries in the associative array for each of the UI objects {the associative array(s) contain(s) global context flags to tell the system the status of the programs/applications/folders, based on the given information, the system will provide or turn on/off some of the necessary options to the user (figs. 4-5)); and

presenting zero or more of the plurality of UI objects as part of the UI to the end-user based upon a result of the Boolean comparison (e.g., figs.4-9).

As to claim 2, Giesen shows the method wherein the step of presenting to an end-user the at least one UI object further comprising at least one of the following:

presenting a visual image (graphical menu, e.g., figs. 4-9);

creating a vibration;

playing a sound;

emitting a scent; and

emitting a taste.

As to claim 3, Giesen shows the method wherein the step of associating an associative array with one or more entries to each of a plurality of UI objects that may be presented as part of a UI includes associating an associative array with zero or more entries to each of a plurality of UI objects that may be presented as part of the UI by retrieving the associative array from a file when an application using the UI is loading (e.g., figs. 4-9).

As to claim 4, Giesen shows the method wherein the global context flags are changed after the application using the UI receives any input from:

an end-user comprising an input from a group of input devices consisting of a keyboard, mouse, pointing device, digitizing pen, light pen, track ball, touch screen, motion detector, chemical sensor, sound sensor, and eye movement sensor (printers, keyboard, etc. e.g., fig. 2).

As to claim 5, Giesen shows the method wherein the global context flags are changed after the application using the UI receives an event-based trigger from one or the following events:

a time event comprising an input from a time of day or elapsed time (the system inherently shows this feature because the invention utilizes a regular IBM computer with Print command of fig. 2 is also available on the menu; it means that the Print command contains well known features such as: a file status, printing status, date, time, etc.); and

an information processing system event comprising a file status, a printing status, a modem status or a power supply status (the Print command contains well known features such as: a file status, printing status, date, time, etc.).

As to claim 6, Giesen shows the method wherein the step of performing a Boolean operation is selected from the group of Boolean operations consisting of “AND”, “OR” and “XOR” {the system will compare the state of user selection, and the associative array(s) contain(s) global context flags to tell the system the status of the programs/applications/folders, based on the given information, the system will provide or turn on/off some of the necessary options to the user (figs. 4-5)}.

As to claim 7, Giesen shows the method wherein the step of associating one or more attributes to each of a plurality of UI objects from a table so that any changes in the table can be made without recompiling and/or re-linking an application using the UI (menus options of figs. 4-9 are also tables that the items can be changed/removed/deleted without recompiling the application).

As to claims 8-10, they are the equivalent system claims of method claims 1, 4, and 2 respectively and are rejected under a similar rationale.

As to claims 11-17, they are the equivalent program product claims of method claims 1-7 respectively and are rejected under a similar rationale.

As to claim 18, Giesen shows a method for presenting User Interface (UI) objects as pad of a UI on an information processing system, the method comprising:

creating a first associative array for controlling a property of at least one UI object in a UI {different groups of the folders associate with an appropriate menu for that group (figs. 4-9 show different menu options or some of the options is removed. It means associative array(s) contain(s) global context flags to tell the system the status of the programs/applications/folders,

Art Unit: 2179

based on the given information, the system will provide or turn on/off some of the necessary options to the user (figs. 4-9));

performing a Boolean comparison between:

at least one global context flag in the first associative array (An associative array of the system is an array of listing states causing the system whether to dim (grey out) or remove options from menu options, e.g., figs. 2-5, and the different menu options between figs. 4 and 5); and

at least one entry in a second associative array of at least one UI object graphical selections associated with the at least one UI object (the different menu options between figs. 4 and 5); and

presenting updates to the UI objects as part of the UI to an end-user in response to the Boolean comparison {the system will compare the state of user selection, and the associative array(s) contain(s) global context flags to tell the system the status of the programs/applications/folders, based on the given information, the system will provide or turn on/off some of the necessary options to the user (figs. 4-5)},

wherein the first associative array and the second associative array are indexed by a string (associative arrays as explained above, e.g., figs. 4-9).

As to claim 19, Giesen shows the method further comprising:

receiving at least one of:

a response from the end-user to the updates to the UI objects (figs. 4-5 show different menu options or some of the options is grey out. It means associative array(s) contain(s) global context flags to tell the system the status of the programs/applications/folders, based on the given

Art Unit: 2179

information, the system will provide or turn on/off some of the necessary options to the user (figs. 4-5)), and an event-based trigger.

As to claim 20, Giesen shows the method further comprising:

changing a state of the global context flag based on the response received from the end-user (the user make sections and interact with the menu items; based on user selections, figs. 4-5 show different menu options or some of the options is grey out).

As to claims 21-23, they are the equivalent method claims of claims 4-6 respectively and are rejected under a similar rationale.

### ***Response to Arguments***

3. Applicant's arguments filed 08/07/06 have been fully considered but they are not persuasive.

Applicant has argued and Examiner disagrees with the following reasons:

**I. Microsoft Explore Screen Captures ("Microsoft ESC"):**

*a. There are no associative arrays, global context flag, or comparison between the global context flag and one or more entries in the associative array, and the choices from Microsoft ESC are not pre-selected by a programmer to be available are grayed out, etc.*

Microsoft MSC clearly provides the window with two panes (figs. 1-4), which contain folders, documents, programs, applications, software, etc., and different groups of the folders associate with an appropriate menu for that group (figs. 2-4 show different menu options or some of the options is grey out). It clearly means associative array(s) contain(s) global context flags, which must be

programmed/setup and existed in the back ground of the OS by the programmers, to be able to tell the OS the status of the programs/applications/folders, based on the given information, the OS will provide or turn on/off some of the necessary options to the user (figs. 2-5). It is the well-known concept of programming for any OS or applications to provide only the appropriate menus/options/icons to prevent confusion and improve user-friendly purposes.

*b. Microsoft ESC does not teach Boolean comparison of two associative arrays.*

Microsoft MSC shows the associative array(s) contain(s) global context flags, which must be programmed/setup and existed in the back ground of the OS by the programmers, to be able to tell the OS the status of the programs/applications/folders, based on the given information, the OS will provide or turn on/off some of the necessary options to the user (figs. 2-5). It means the Microsoft MSC does not only compare two associative arrays by Boolean comparison (yes/no, on/off) but also to possibly compare with other type arrays to determine the appropriate output menus/fields/options back to the user.

**II. Giesen et al. ("Giesen", U.S. Patent No. 6,826,729 B1).**

*c. There are no associative arrays, global context flags, and comparison between the global context flag and one or more entries in the associative arrays.*

Giesen clearly teaches the associative array is the array of listing states causing the system whether to dim (gray out) or remove options from menu options (e.g., figs. 4-5). The associative array(s) contain(s) global context flags to tell the system the status of the programs/applications/folders, based on the given

information, the system will provide or turn on/off some of the necessary options to the user (figs. 4-5). The predetermined setups cause the system to compare two associative arrays by Boolean comparison (yes/no, on/off), and possibly compare with other type arrays to determine the appropriate output menus/fields/options back to the user.

### ***Conclusion***

4. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Truc T. Chuong whose telephone number is 571-272-4134. The examiner can normally be reached on M-Th and alternate Fridays 8:30 AM - 5:00 PM.

Art Unit: 2179

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on (571) 272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Truc T. Chuong

10/15/06

**BA HUYNH**  
**PRIMARY EXAMINER**